P&S Processing-in-Memory

Real-World Processing-in-Memory Architectures: SK Hynix Accelerator-in-Memory

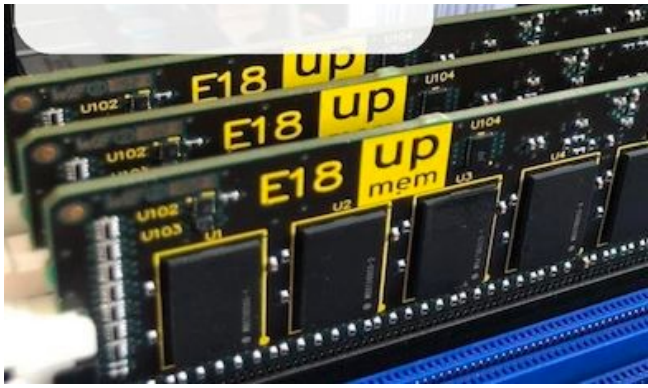
Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich

Fall 2022

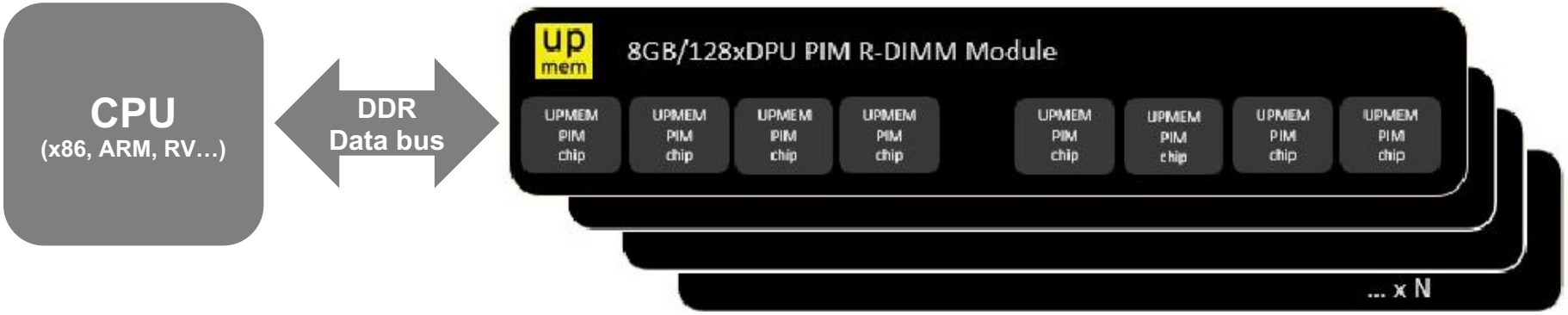
15 November 2022

UPMEM Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.



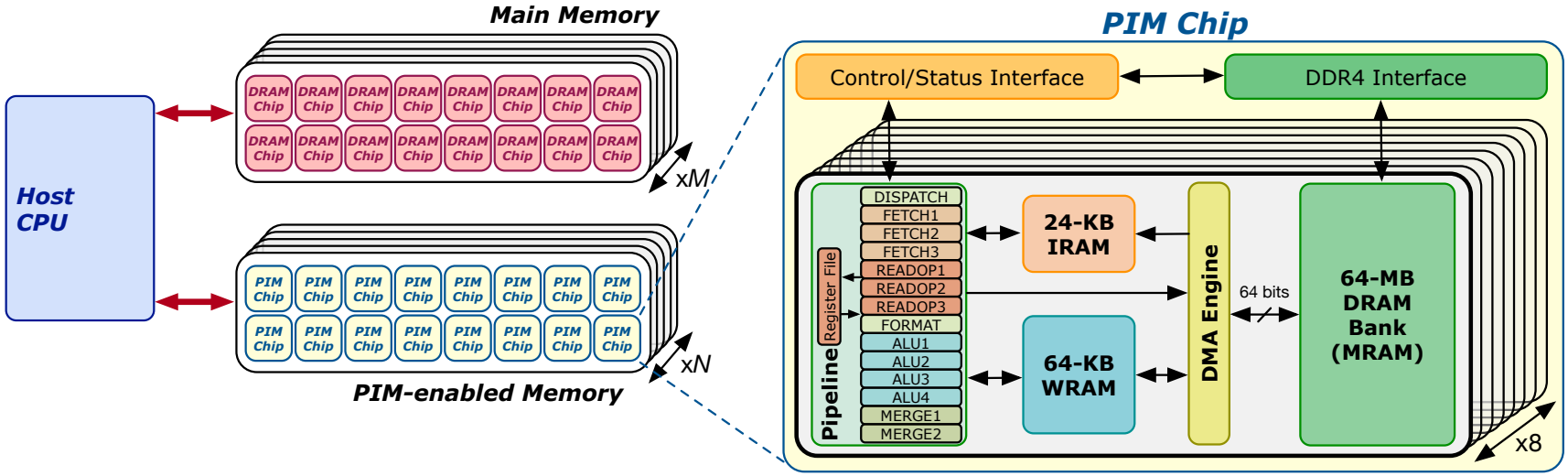
- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth



https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

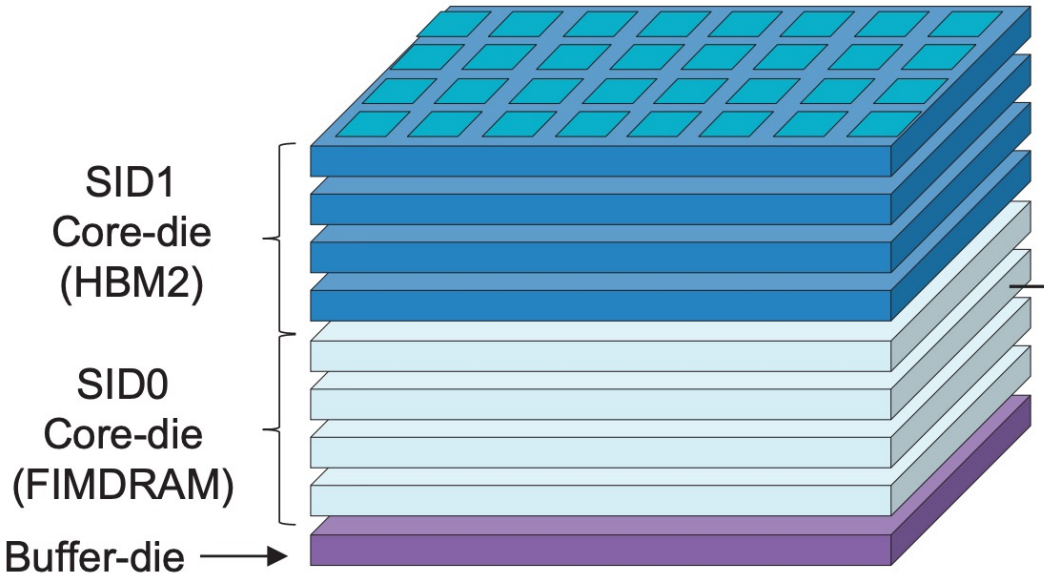
Recall: UPMEM PIM System Organization

- A UPMEM DIMM contains 8 or 16 chips
  - Thus, 1 or 2 ranks of 8 chips each
- Inside each PIM chip there are:
  - 8 64MB banks per chip: Main RAM (MRAM) banks
  - 8 DRAM Processing Units (DPUs) in each chip, 64 DPUs per rank



FIMDRAM: Chip Structure

- FIMDRAM based on HBM2

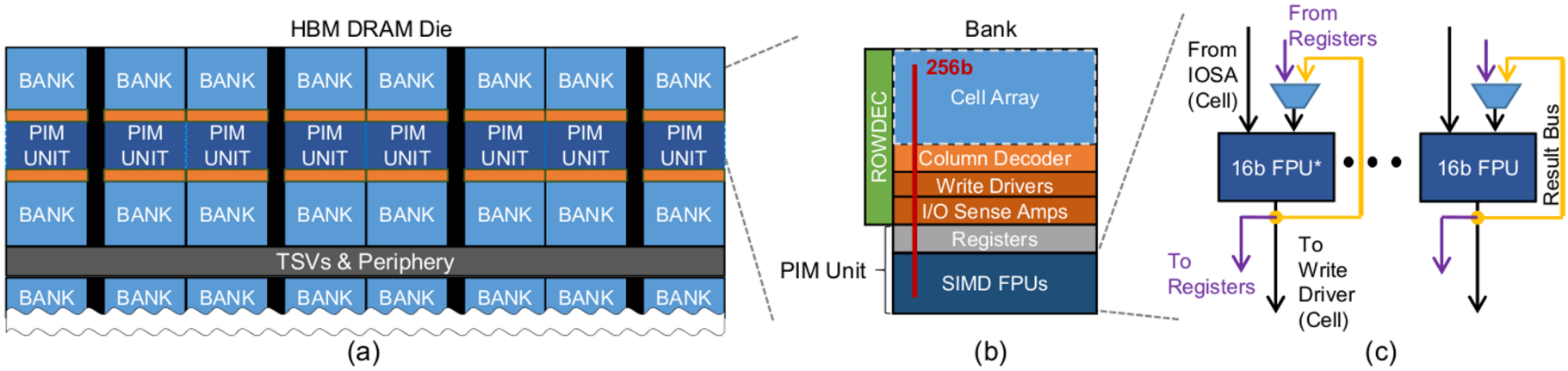


[3D Chip Structure of HBM with FIMDRAM]

Chip Specification
128DQ / 8CH / 16 banks / BL4
32 PCU blocks (1 FIM block/2 banks)
1.2 TFLOPS (4H)
FP16 ADD / Multiply (MUL) / Multiply-Accumulate (MAC) / Multiply-and- Add (MAD)

FIMDRAM: System Organization (III)

- PIM units respond to standard DRAM column commands (RD or WR)
  - Compliant with unmodified JEDEC controllers
- They execute one wide-SIMD operation commanded by a PIM instruction with deterministic latency in a lock-step manner
- A PIM unit can get 16 16-bit operands from IOSAs, a register, and/or the result bus



SK Hynix Accelerator-in-Memory

SK Hynix Accelerator-in-Memory (2022)

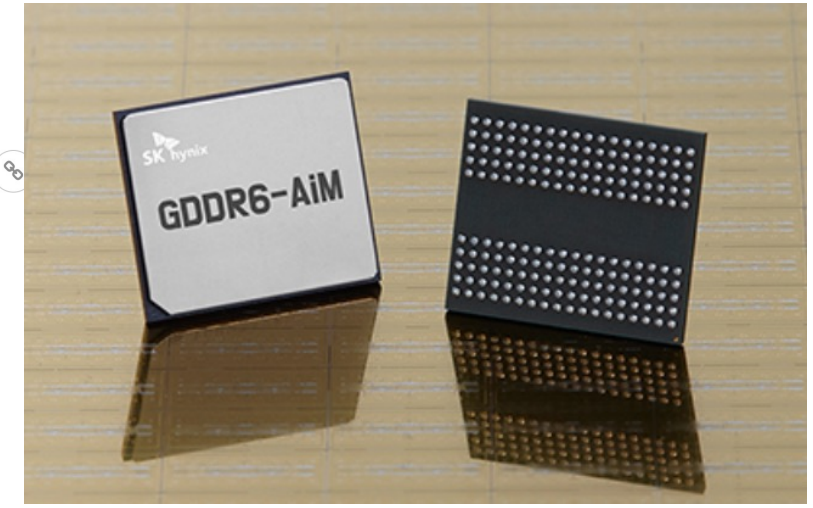
SKhynix NEWSROOM

ENG

INSIGHT | SK hynix STORY | PRESS CENTER | MULTIMEDIA | Search

SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022



Seoul, February 16, 2022

SK hynix (or "the Company", www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

*PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world's most prestigious semiconductor conference, 2022 ISSCC*, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

*ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of "Intelligent Silicon for a Sustainable World"

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AiM (Accelerator* in memory). The GDDR6-AiM adds computational functions to GDDR6* memory chips, which process data at 16Gbps. A combination of GDDR6-AiM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AiM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.

11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

Seongju Lee, SK hynix, Icheon, Korea

In Paper 11.1, SK Hynix describes an 1ynm, GDDR6-based accelerator-in-memory with a command set for deep-learning operation. The 8Gb design achieves a peak throughput of 1TFLOPS with 1GHz MAC operations and supports major activation functions to improve accuracy.

https://news.skhynix.com/sk-hynix-develops-pim-next-generation-ai-accelerator/

Accelerator-in-Memory (ISSCC 2022)

ISSCC 2022 / SESSION 11 / COMPUTE-IN-MEMORY AND

11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

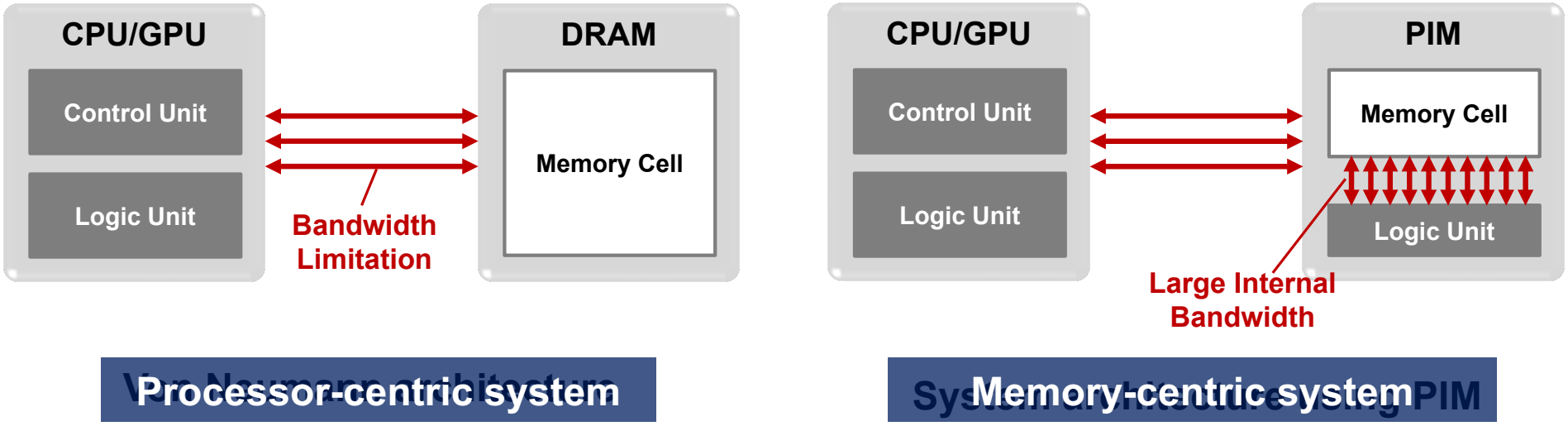
Seongju Lee, Kyuyoung Kim, Sanghoon Oh, Joonhong Park, Gimoon Hong, Dongyoon Ka, Kyudong Hwang, Jeongje Park, Kyeongpil Kang, Jungyeon Kim, Junyeol Jeon, Nahsung Kim, Yongkee Kwon, Kornijcuk Vladimir, Woojae Shin, Jongsoon Won, Minkyu Lee, Hyunha Joo, Haerang Choi, Jaewook Lee, Donguc Ko, Younggun Jun, Keewon Cho, Ilwoong Kim, Choungki Song, Chunseok Jeong, Daehan Kwon, Jieun Jang, Il Park, Junhyun Chun, Joohwan Cho

SK hynix, Icheon, Korea

https://doi.org/10.1109/ISSCC42614.2022.9731711

AiM: Exploiting Bank Parallelism

- Memory bandwidth is not enough for many ML workloads



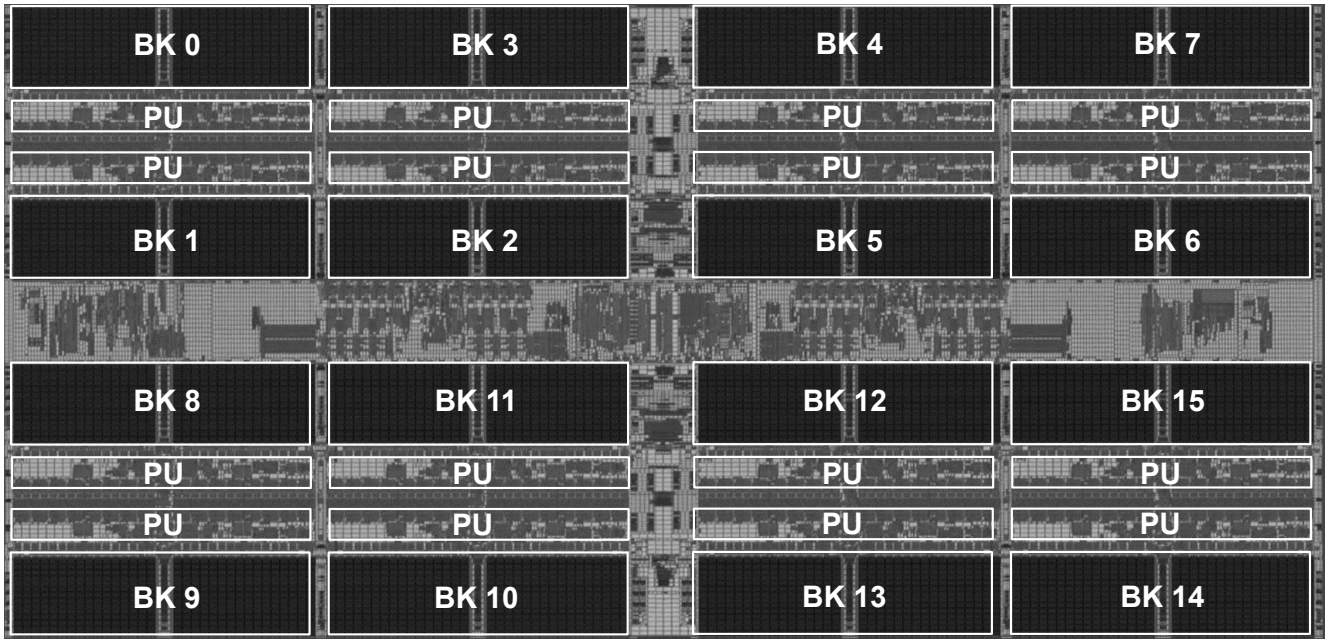
Onur Mutlu, "Memory-Centric Computing", Keynote Talk at the Thoughtworks Engineering for Research Symposium (E4R), Virtual, 19 February 2022.

Lee et al., A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications, ISSCC 2022

AiM: Chip Implementation

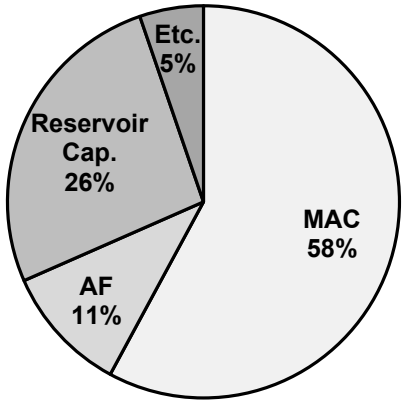
- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph



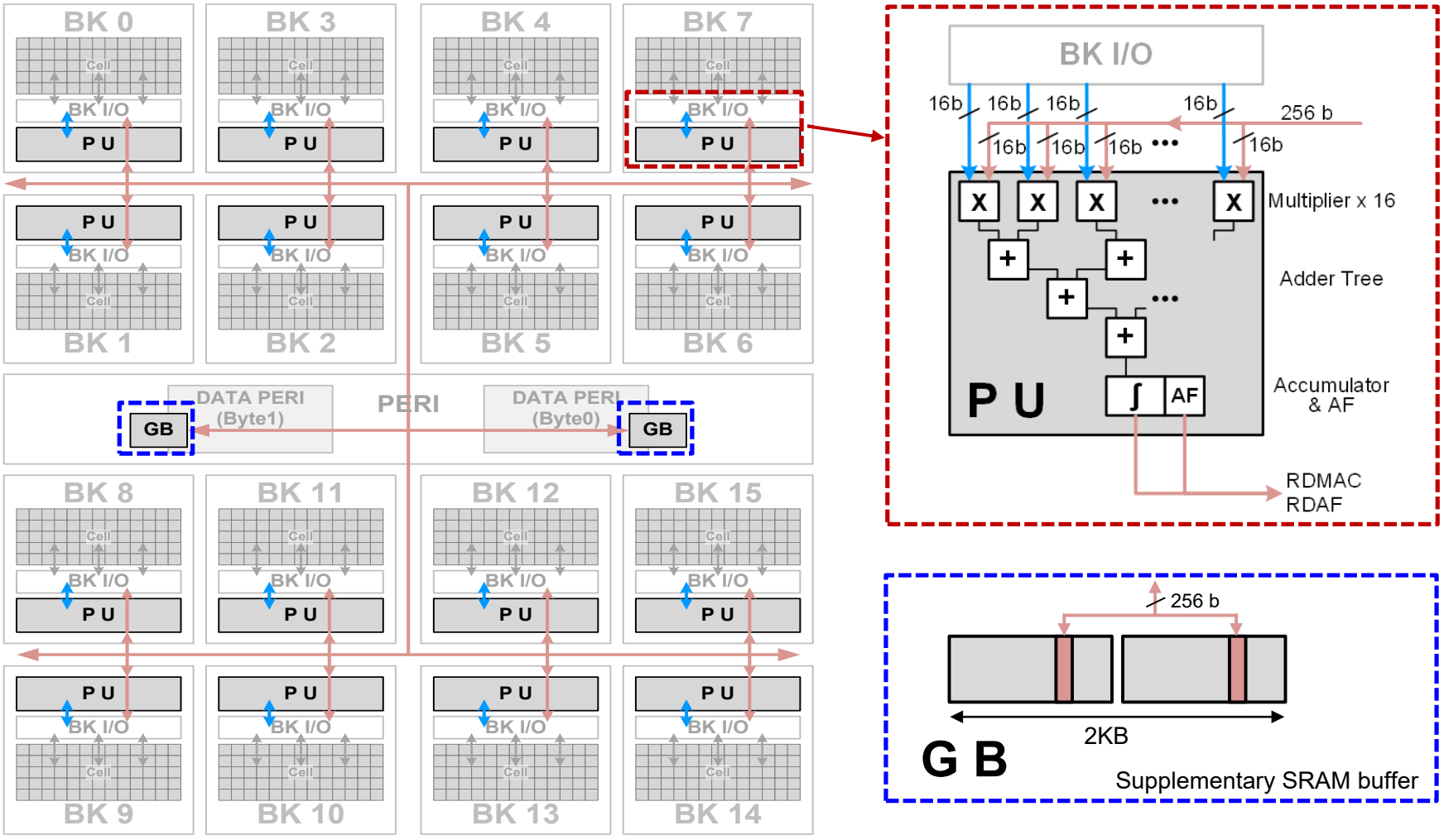
1 Process Unit (PU) Area

Total	0.19mm^2
MAC	0.11mm^2
Activation Function (AF)	0.02mm^2
Reservoir Cap.	0.05mm^2
Etc.	0.01mm^2



AiM: System Organization

- GDDR6-based AiM architecture



AiM Commands

AiM: Command Set

- New commands for computation

Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

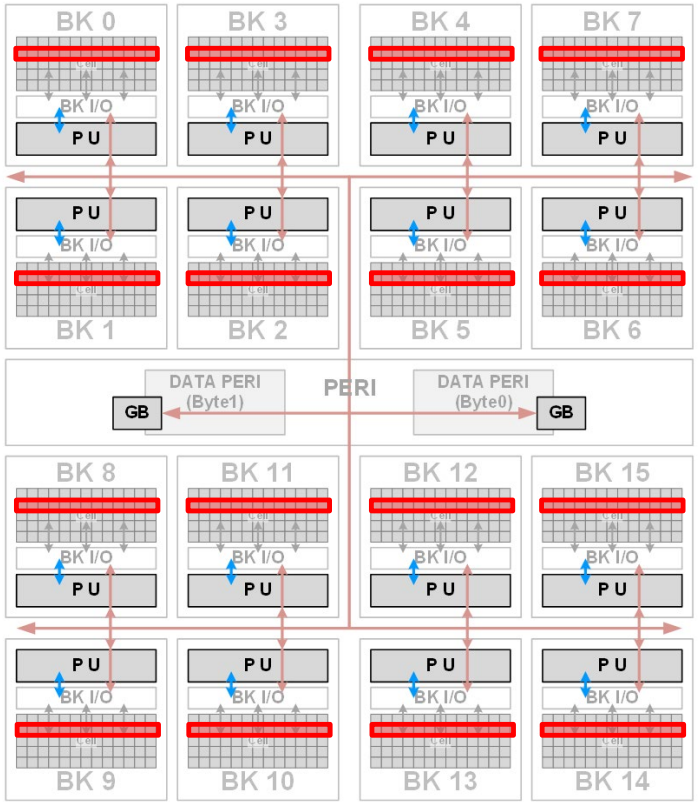
AiM: Command Set: ACT4, ACT16

- Activate 4 or 16 banks at once

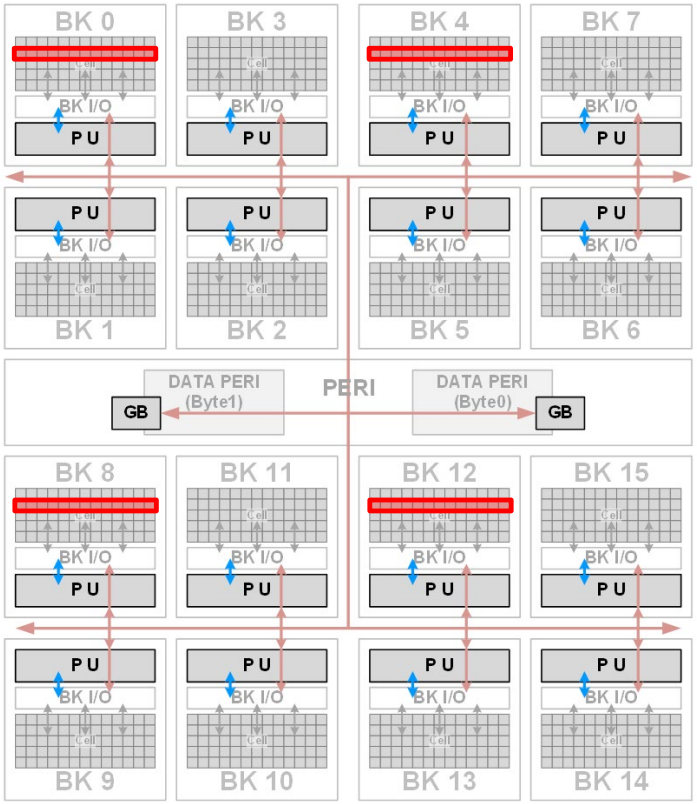
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

Activated Word Line

16-bank Active



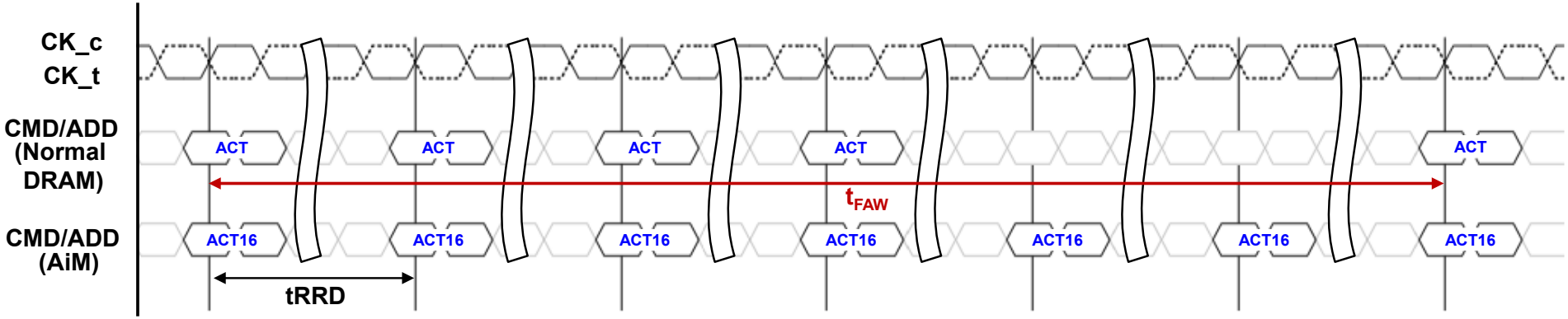
4-bank Active



AiM: Four Active Window (t_{FAW})

- ACT16 operations are possible without t_{FAW} constraint
  - In normal DRAM only 4 ACT are possible every t_{FAW}
  - Reservoir capacitor in each PU

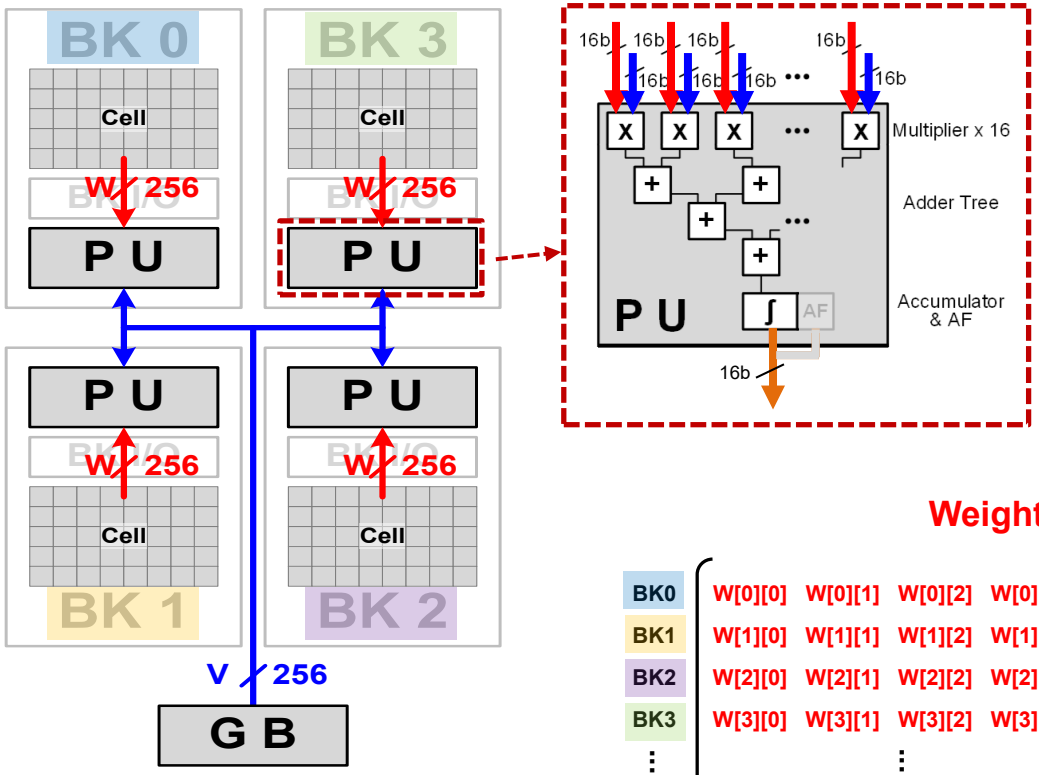
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

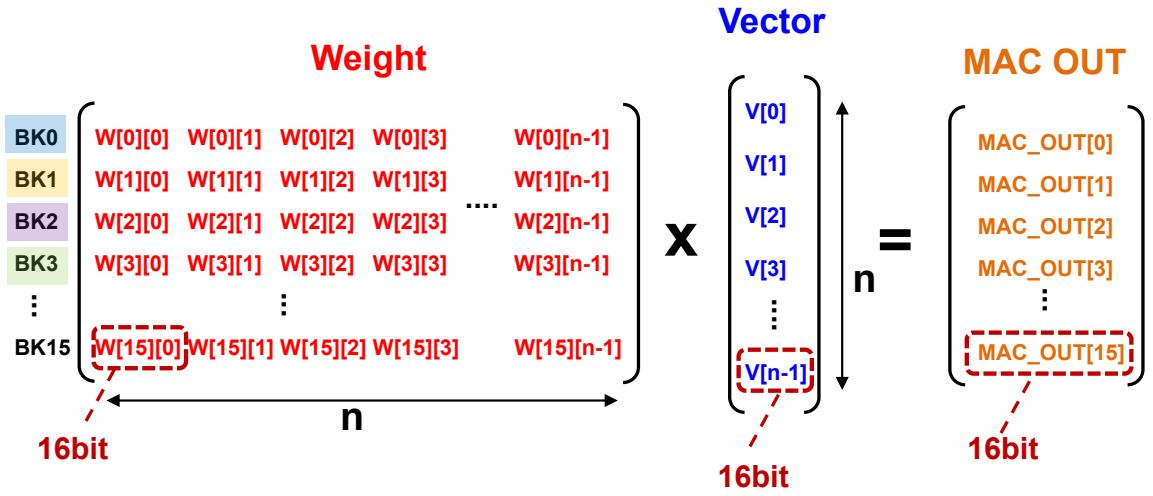


AiM: MAC Operation with Global Buffer

- MAC operation: Weights from the banks, vectors from the GB
- Operates with 16, 4, or 1 PU

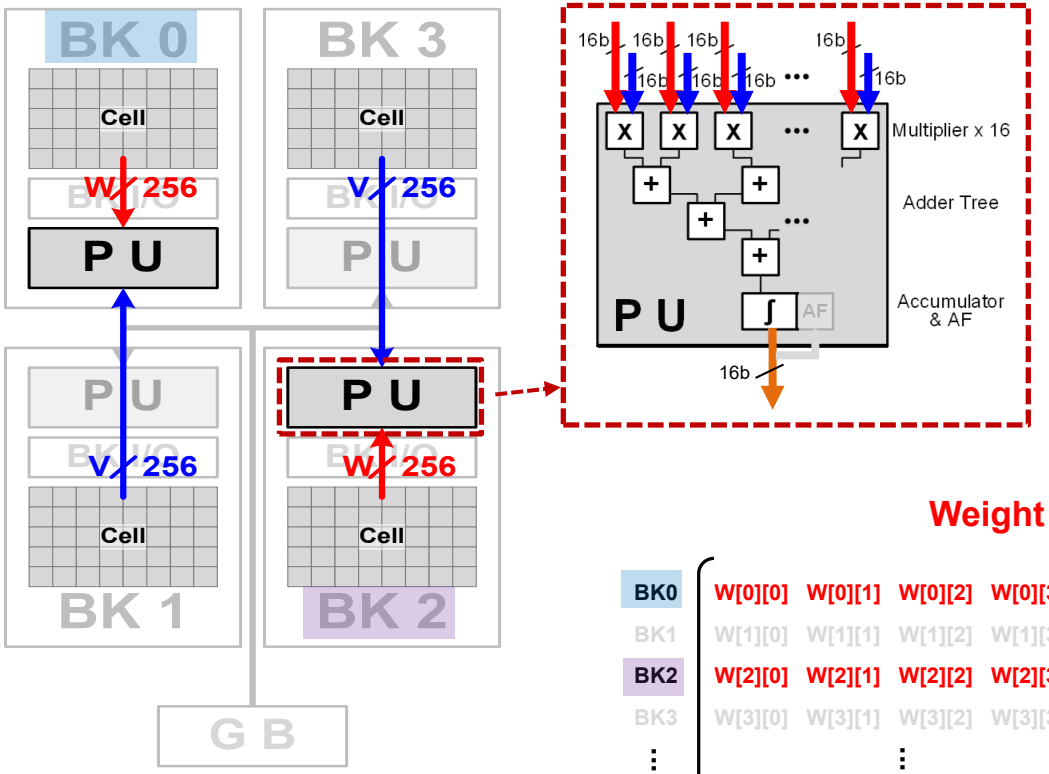
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank



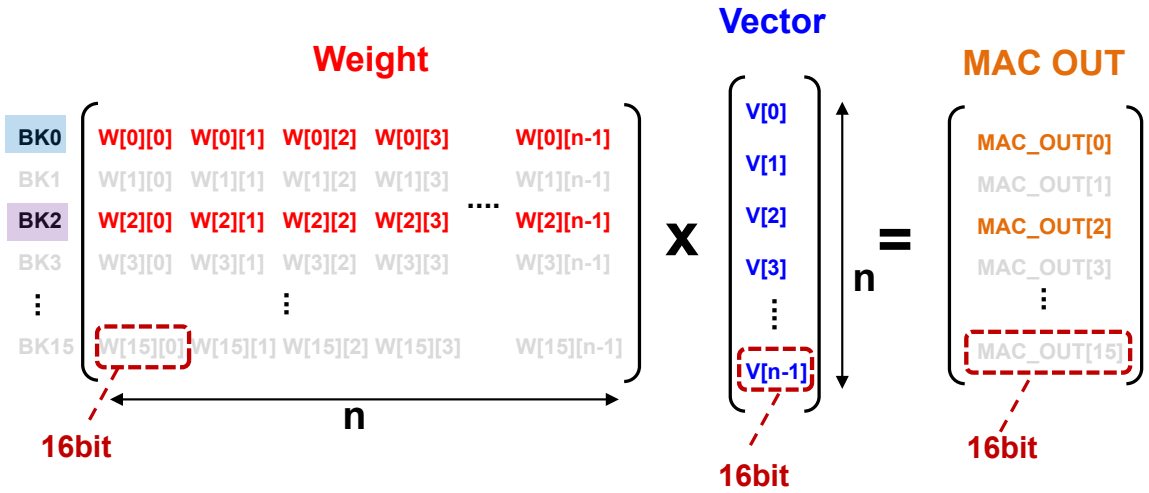


AiM: MAC Operation without Global Buffer

- MAC operation: Weights and vectors from the banks
- Operates with 8, 4, or 1 PU

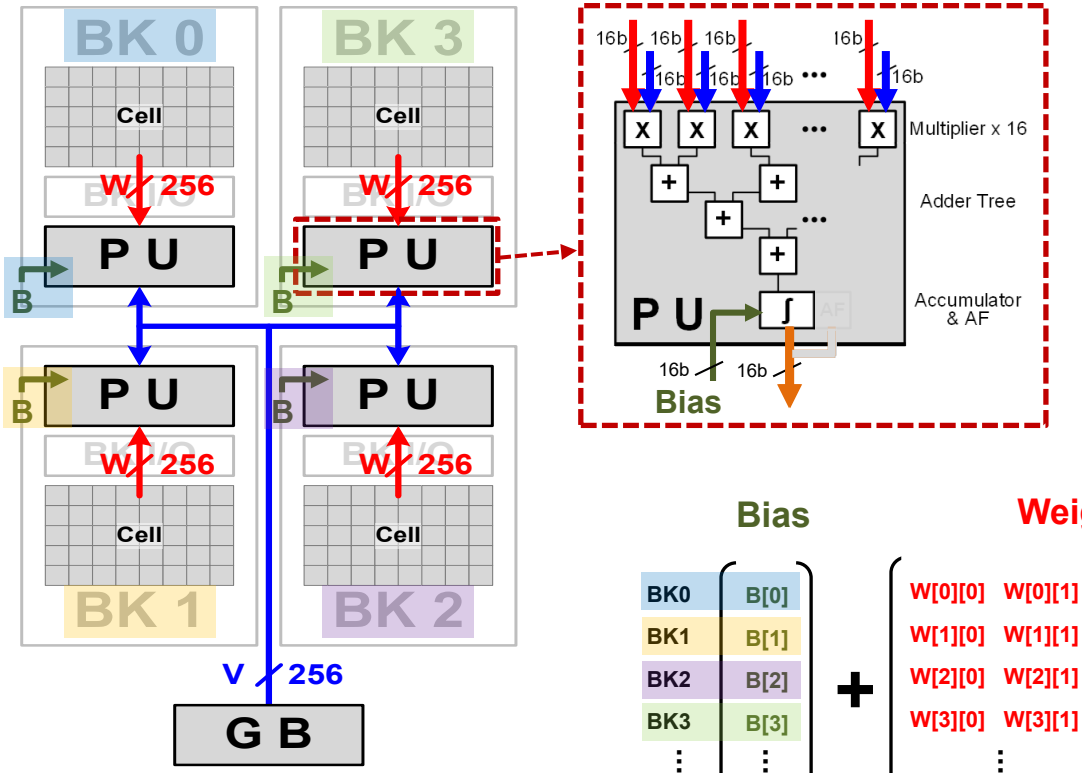


Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

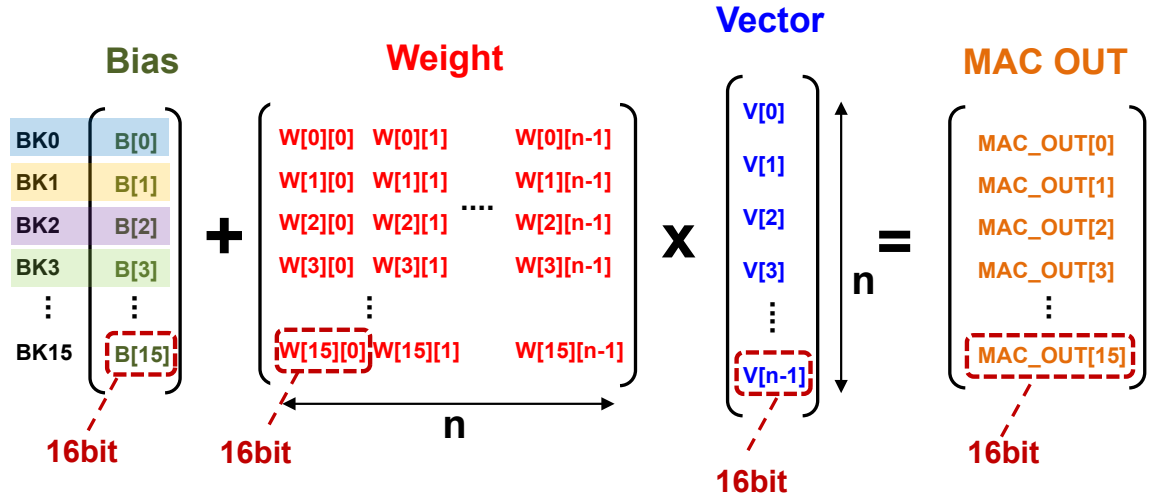


AiM: Write Bias

- Biases can be added to MAC results
  - Different biases in 16 banks at the same time



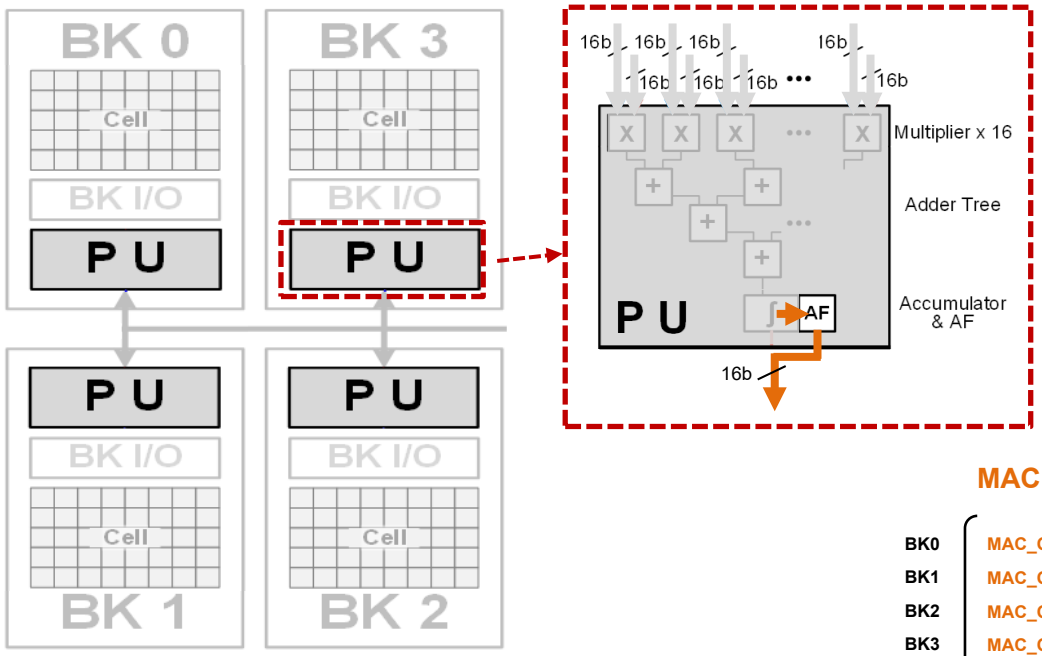
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

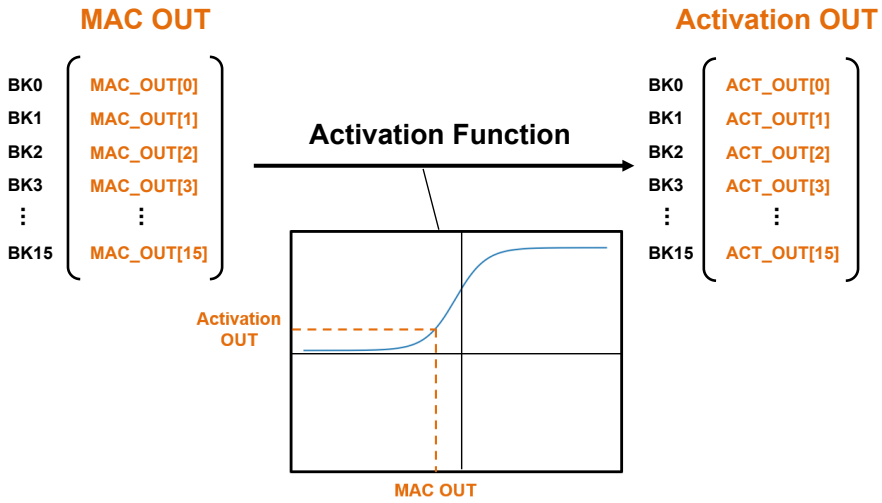


AiM: Activation Function

- Activation function (AF): MAC result as input

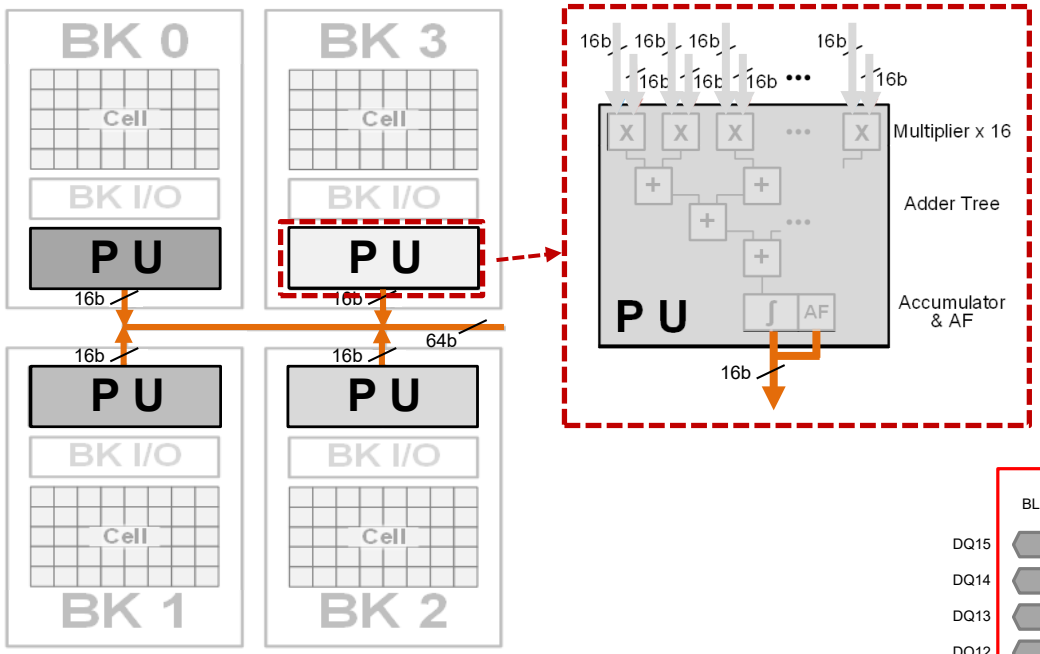
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank



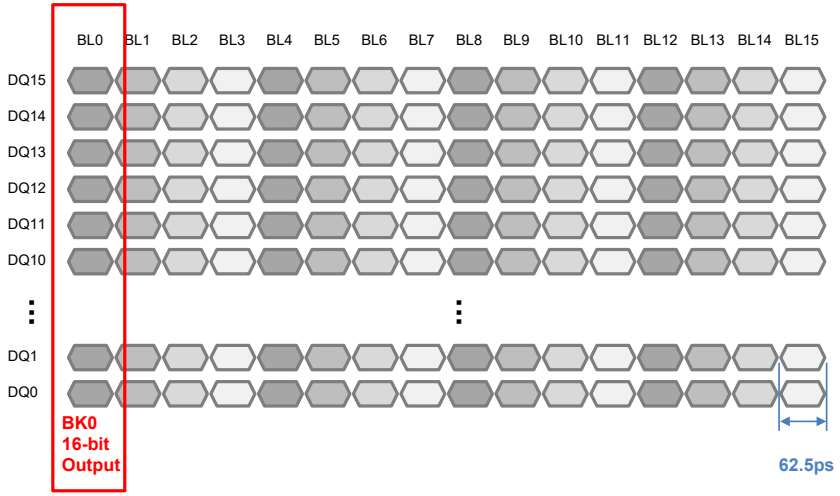


AiM: Read MAC / Read AF

- Read outside DRAM
  - 16 banks x 16 bits = 256 bits

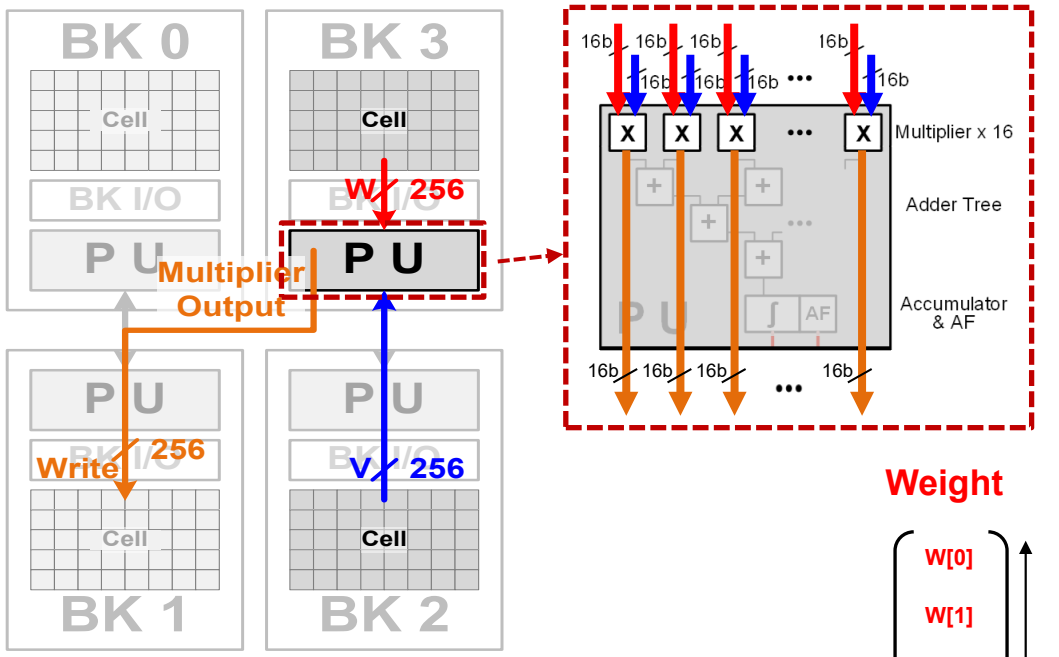


Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

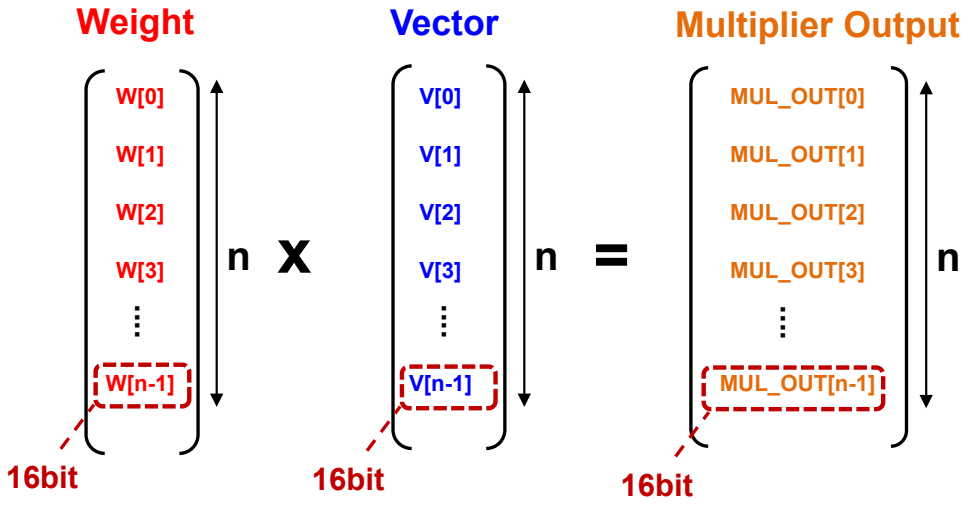


AiM: Element-wise Multiplication

- Multiplies weights and vectors
  - The adder tree is disabled



Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

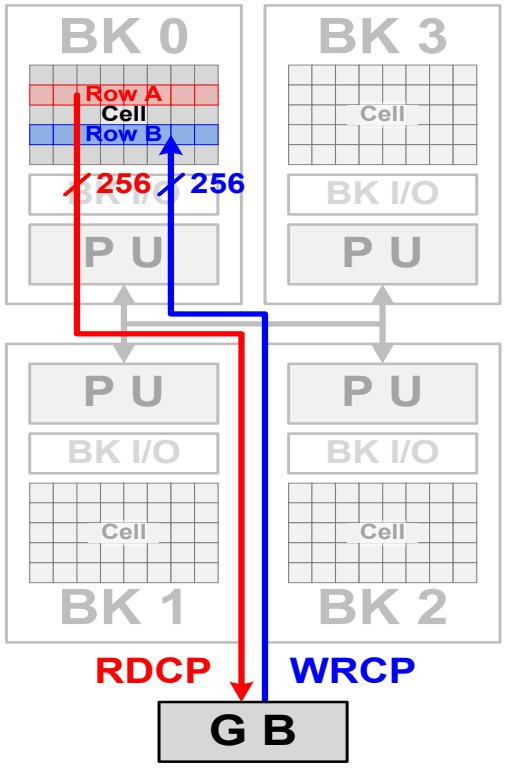


AiM: Copy Operation

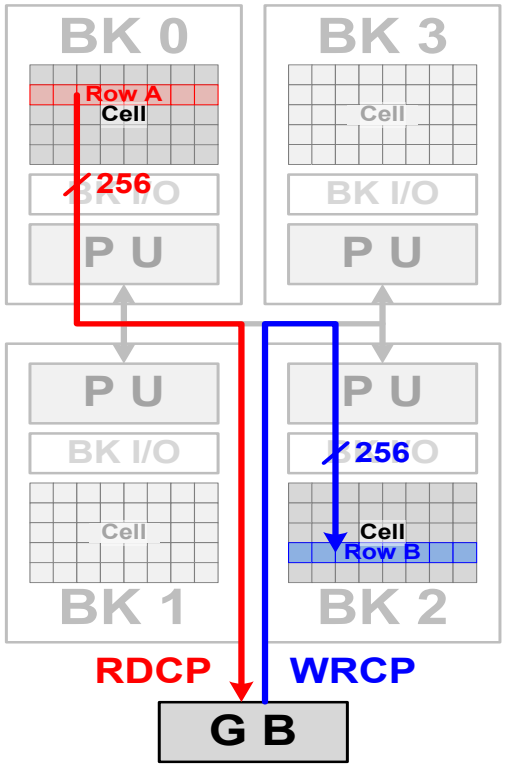
- Copy operation using GB as a intermediate buffer
  - It copies 2 KB from one row to another row

Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

Copy to Same Bank

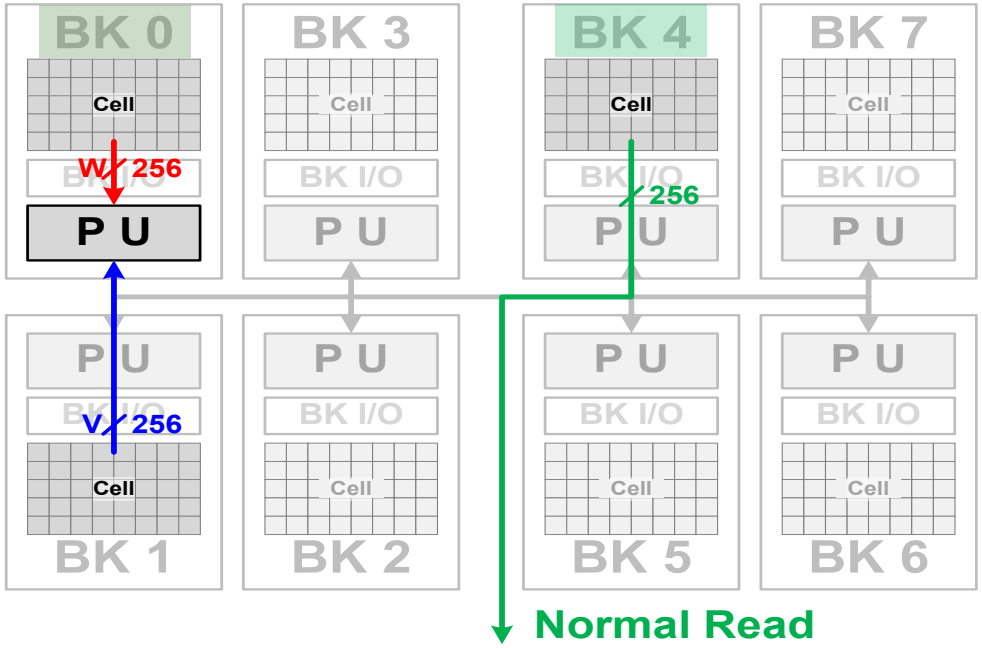


Copy to Another Bank

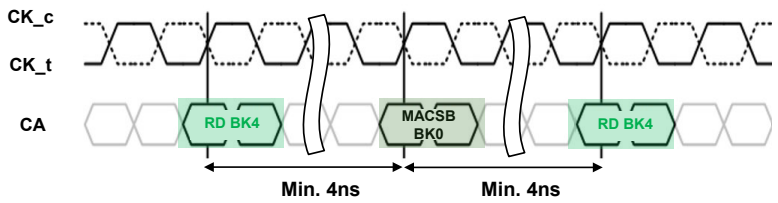


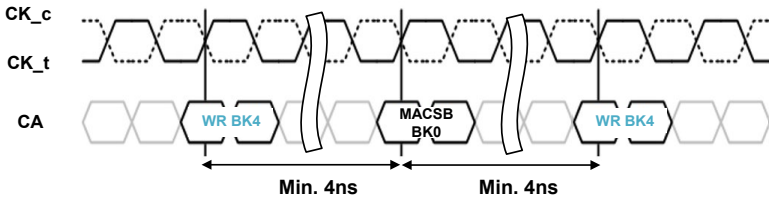
AiM: Simultaneous Computation and Access

- Computation can be performed during normal read/write



Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

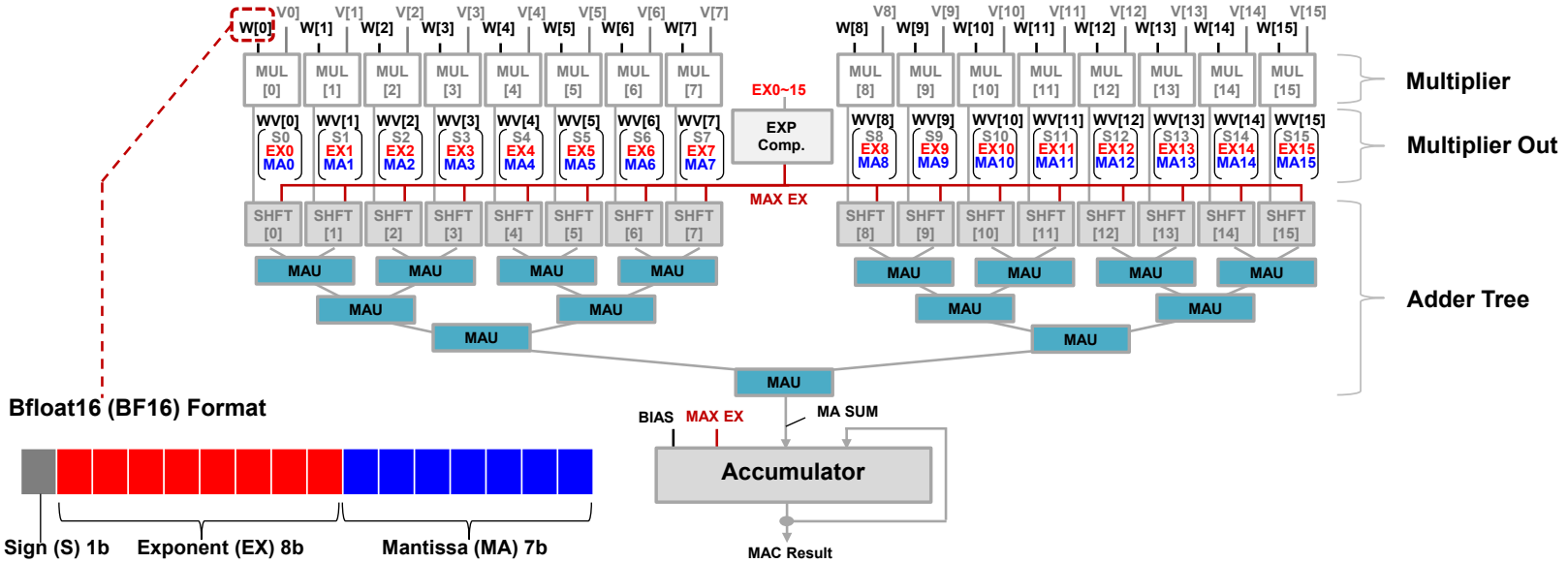


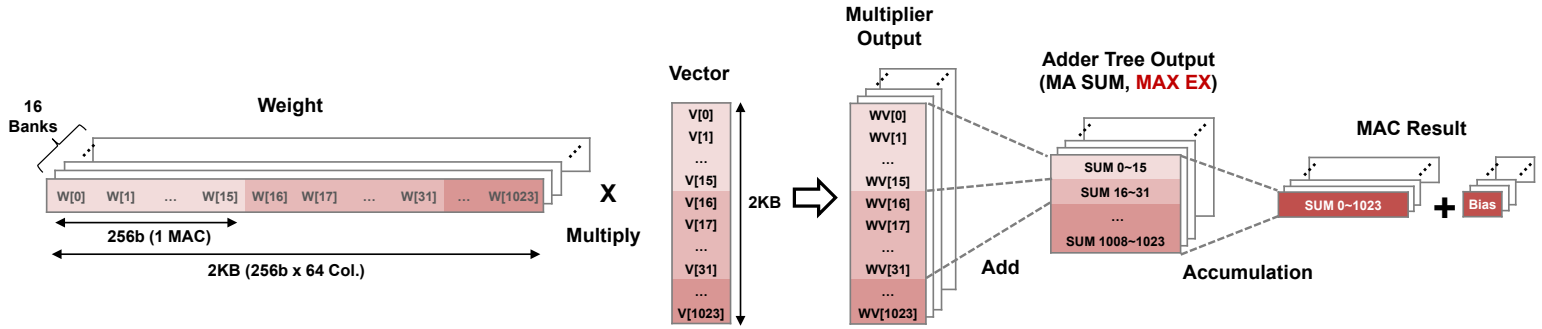


AiM Microarchitecture

AiM: MAC Circuit

- 16 multipliers, adder tree, and accumulator
  - Bfloat16 (BF16) format

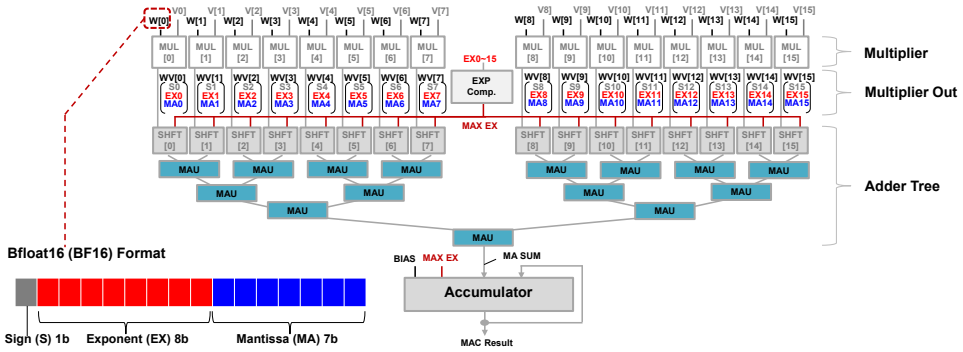


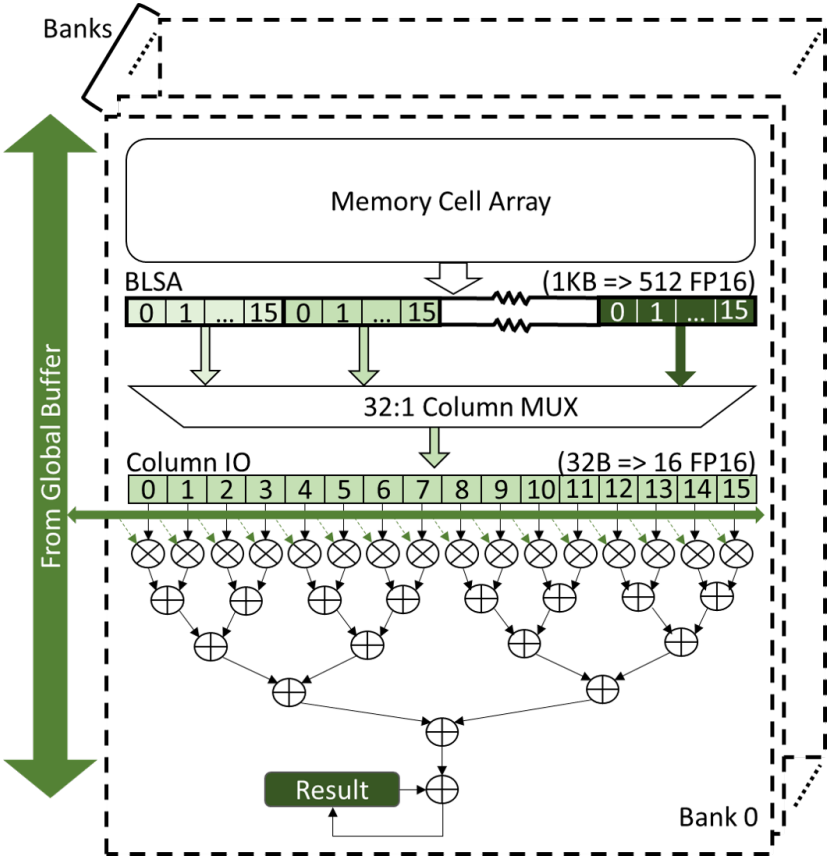


Lee et al., A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications, ISSCC 2022
Kalamkar et al., A Study of BFLOAT16 for Deep Learning Training, arXiv (pre-print), id. 1905.12322, 2019

AiM: MAC Circuit: Prior Work (I)

- 16 multipliers, adder tree, and accumulator

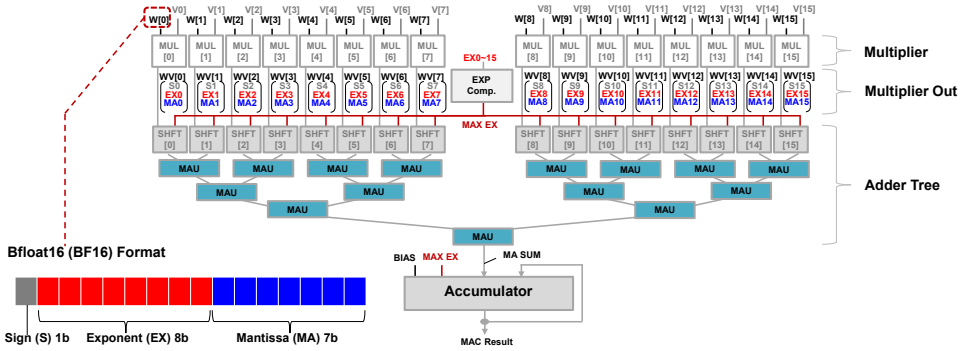


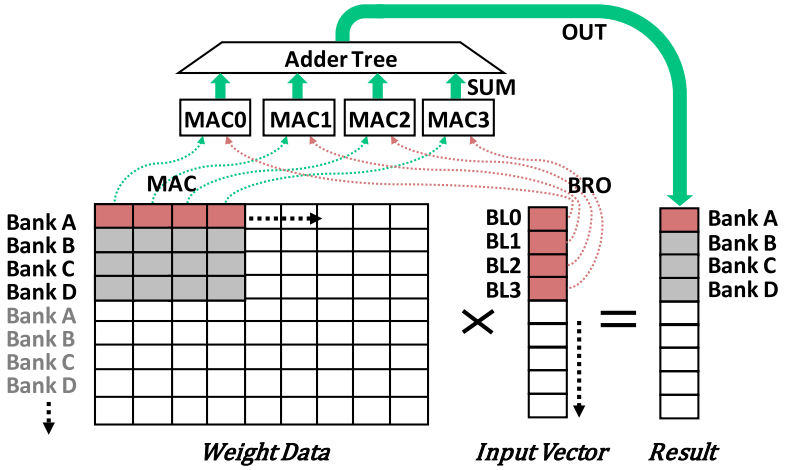


He et al, "Newton: A DRAM-maker's accelerator-in-memory (AiM) architecture for machine learning," MICRO 2020

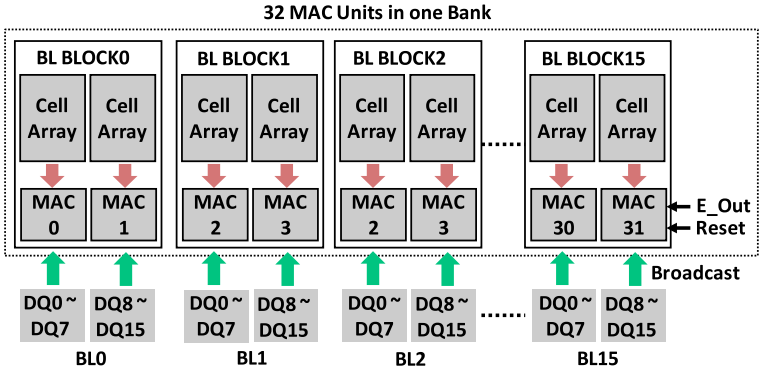
AiM: MAC Circuit: Prior Work (II)

- 16 multipliers, adder tree, and accumulator

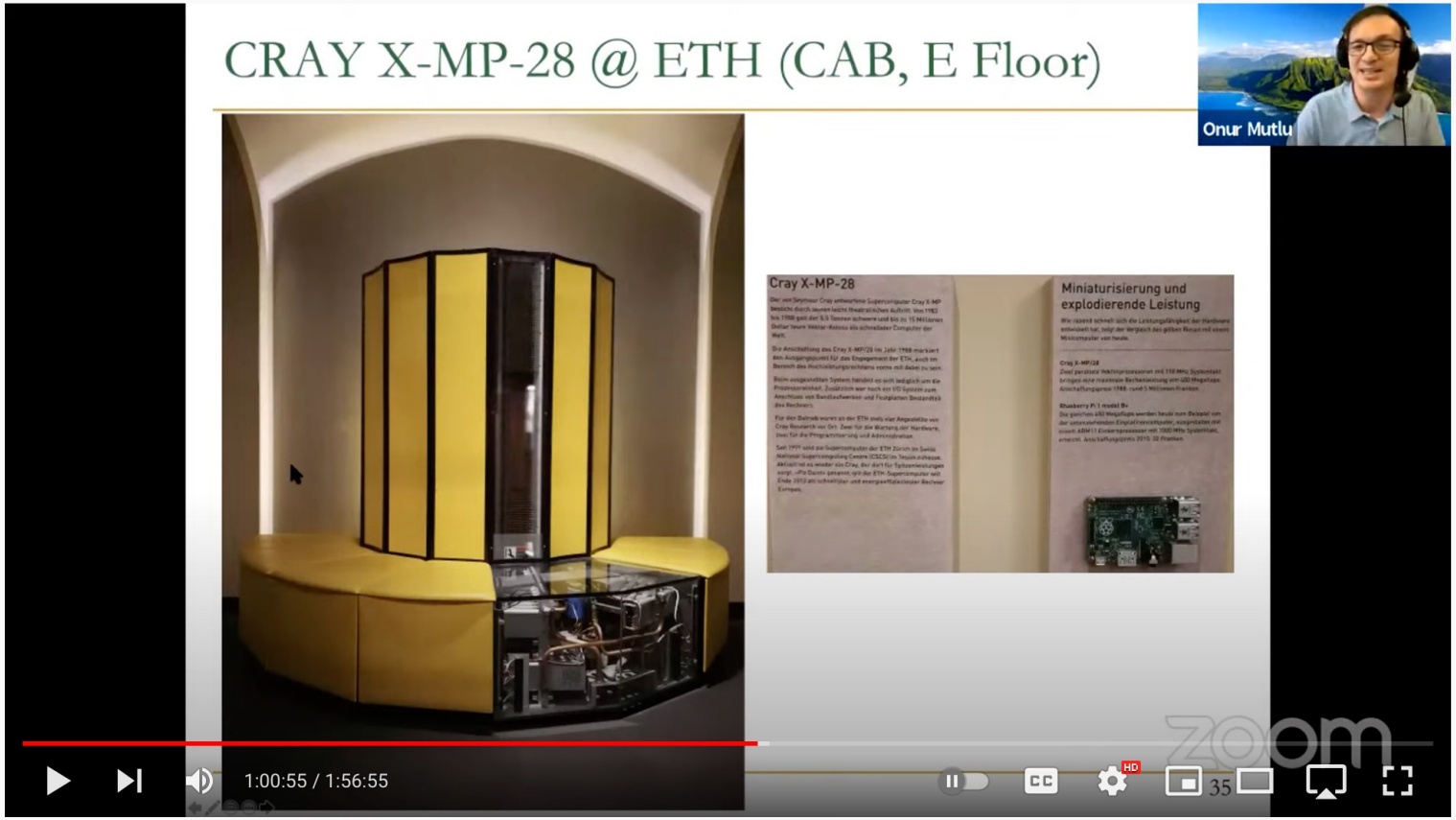




Shin et al, "McDRAM: Low latency and energy-efficient matrix computations in DRAM," IEEE TCADICS (2018)



Lecture on SIMD Processing



DEPARTMENT OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING (D-ITET)

Digital Design & Comp. Arch. - Lecture 20: SIMD Processing (Vector and Array Processors) (Spring'21)

2,677 views • Streamed live on May 14, 2021

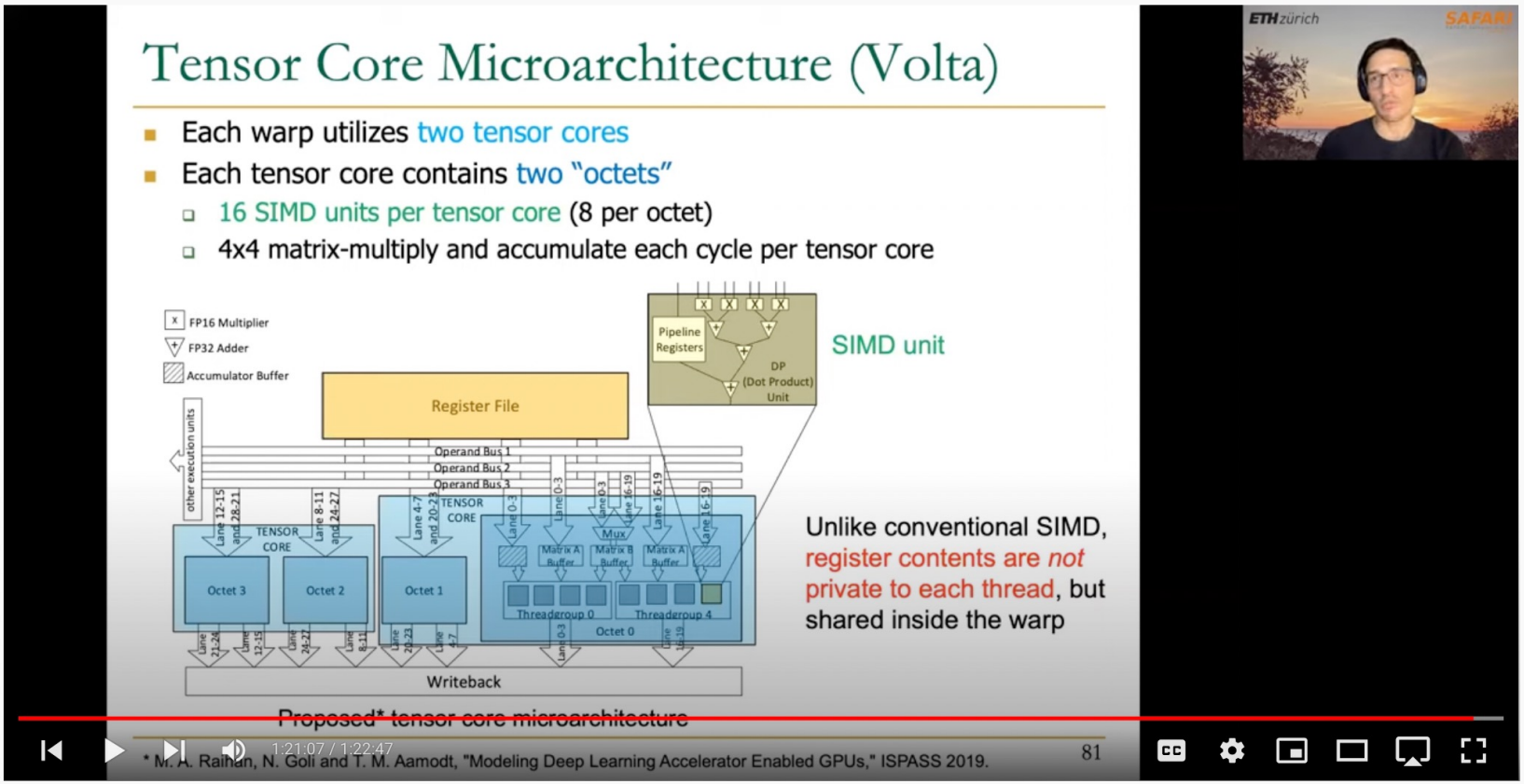
69 1 SHARE SAVE ...

Onur Mutlu Lectures
19.2K subscribers

SUBSCRIBED

https://youtu.be/fP4kZ2Zx_84

Lecture on SIMD Processing and GPUs



HetSys Course: Lecture 2: SIMD Processing and GPUs (Spring 2022)

380 views • Premiered Mar 22, 2022

9 DISLIKE SHARE CLIP SAVE ...

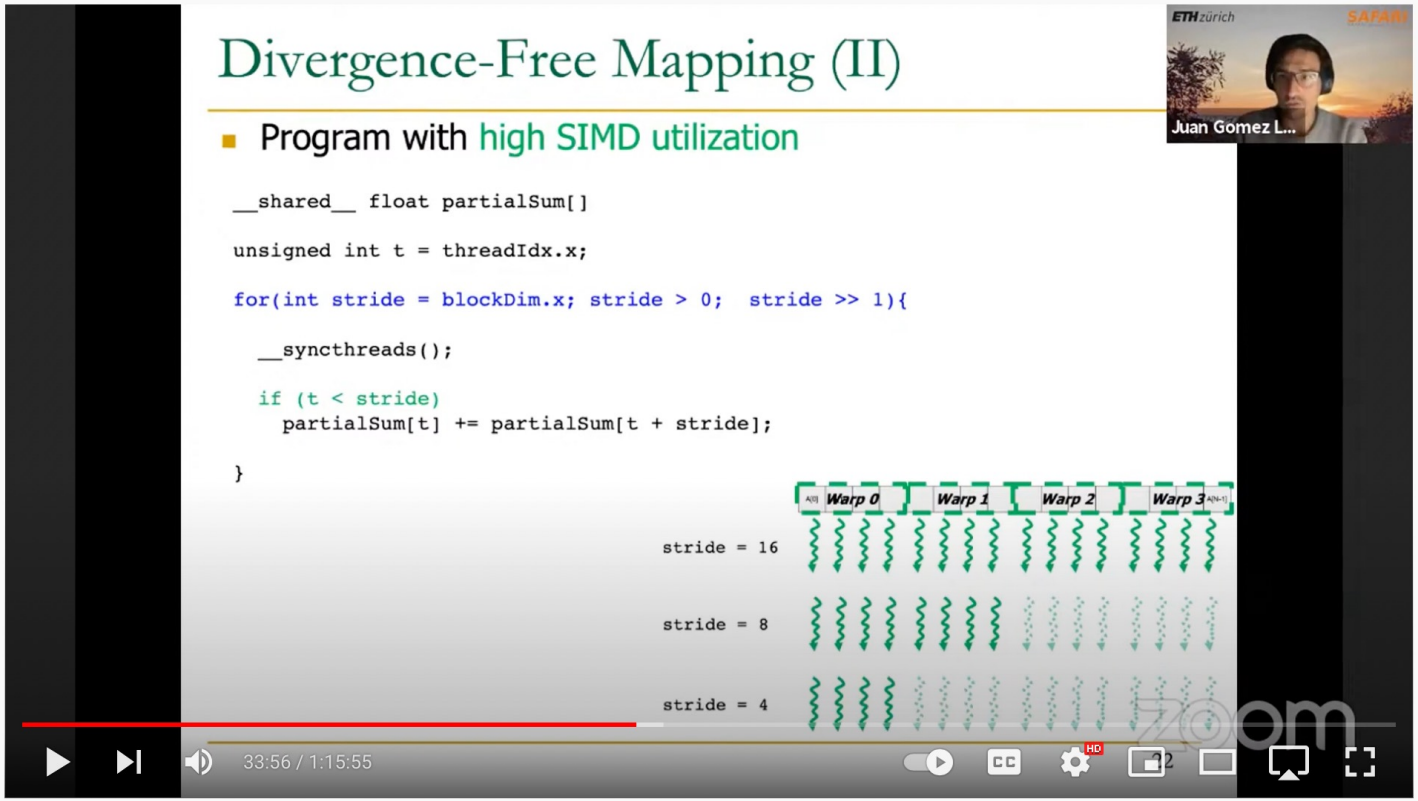
Onur Mutlu Lectures
23.6K subscribers

SUBSCRIBED

Project & Seminar, ETH Zürich, Spring 2022
Hands-on Acceleration on Heterogeneous Computing Systems (
https://safari.ethz.ch/projects_and_s...)

https://youtu.be/hEqk6UMQT0U

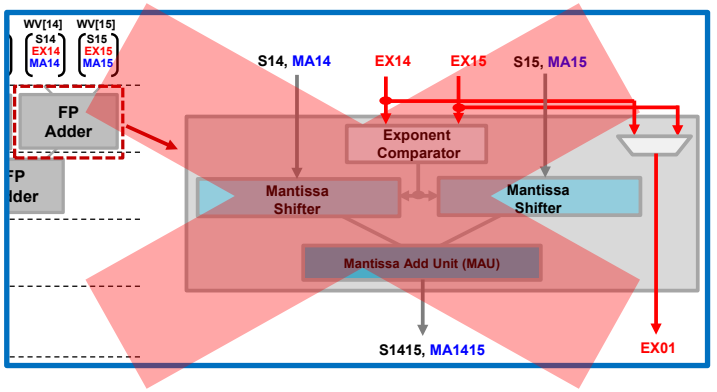
Parallel Reduction on GPU

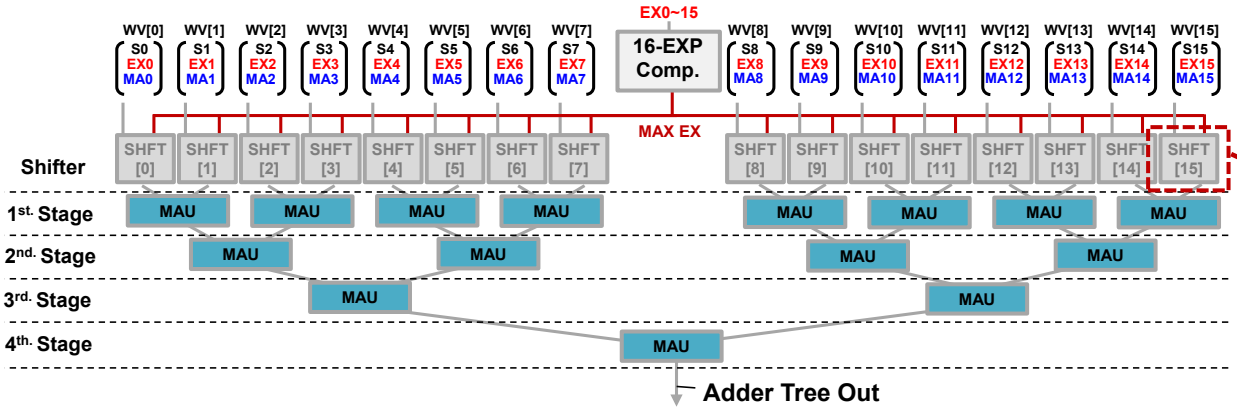


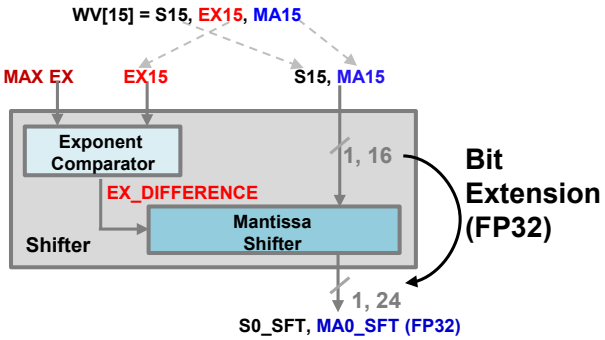
https://youtu.be/PN0h5XPhJIM

AiM: Adder Tree: Bank-wide Mantisa Shift (I)

- Bank-wide Mantisa Shift (BWMS)
  - Find MAX EX of 16 EXs
  - Obtain the differences
  - Shift all MAs by the differences
  - Perform MA additions



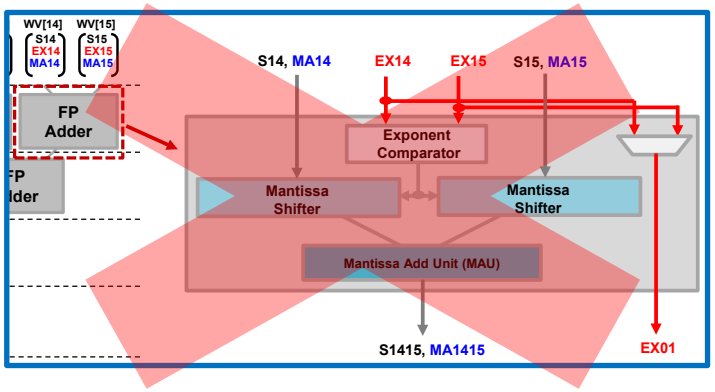


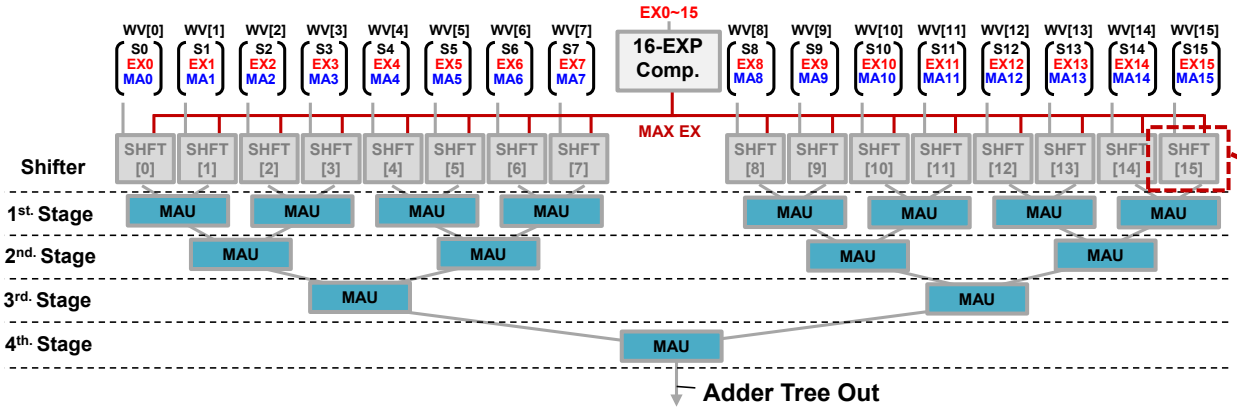


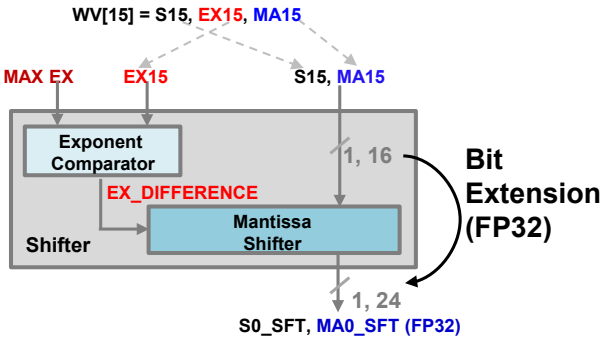
MUL OUT	Before Shifter: Exponent (EX) (8 bits)	Before Shifter: Sign (S), Mantissa (MA) (1, 16 bits)	Diff. w/ Max EX	After Shifter: Exponent (EX) (8 bits)	After Shifter: Sign(S), Mantissa(MA) (1, 24 bits)
WV[0]	00001100	+1.111110011111011	3	00001111	+0.001111110011111101100000 (3b Sft.)
WV[1]	00001111	+1.000000011111011	Max EX		+1.000000011111011000000000
WV[2]	00000100	- 1.111110000000000	11		- 1.111111111111110000000 (11b Sft.)
⋮	⋮	⋮	⋮		⋮
WV[15]	00001000	+1.000000011111111	7		+0.000000100000001111111110 (7b Sft.)

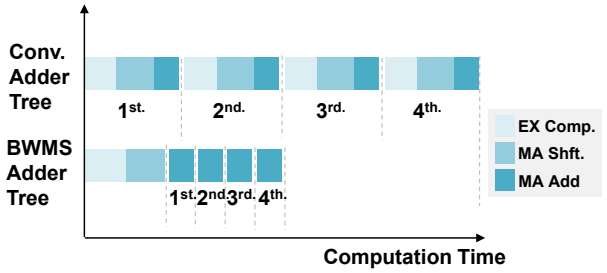
AiM: Adder Tree: Bank-wide Mantisa Shift (II)

- Bank-wide Mantisa Shift (BWMS)
  - Find MAX EX of 16 EXs
  - Obtain the differences
  - Shift all MAs by the differences
  - Perform MA additions





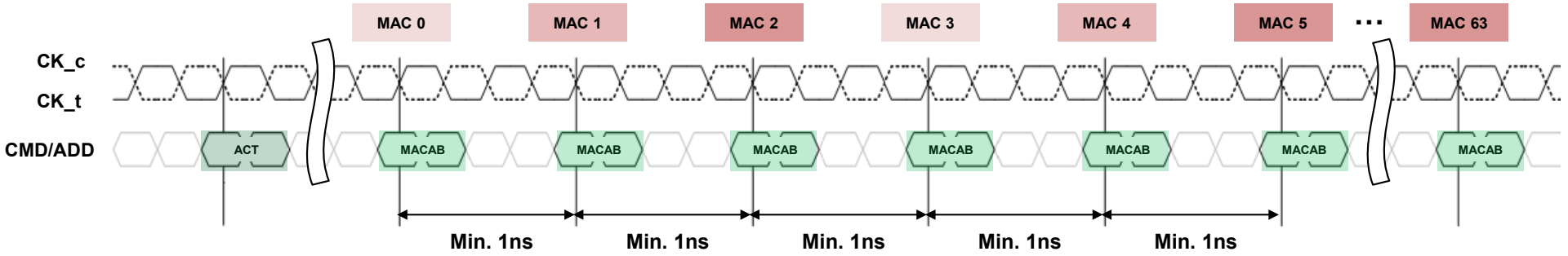




AiM: Accumulation Operation

- Up to 64 MAC commands between ACT
  - 1 MAC every 1ns (t_{CCDS})
  - 64 x 256 bits = 2 KB (row size)

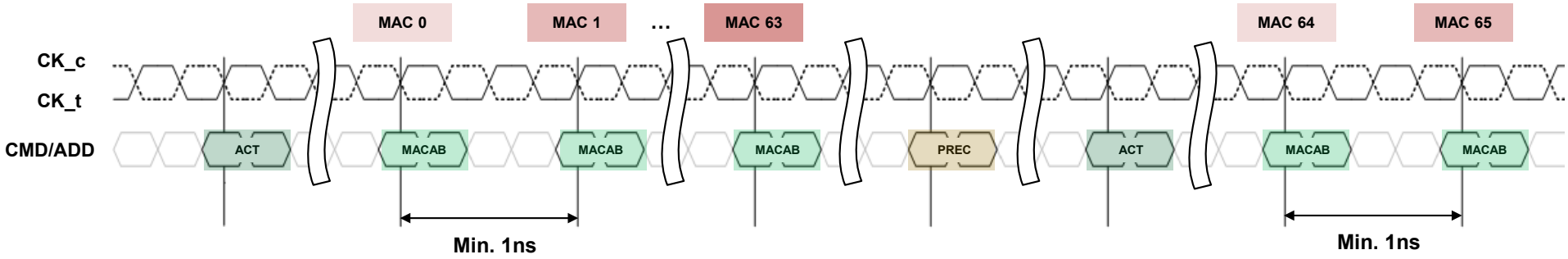
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

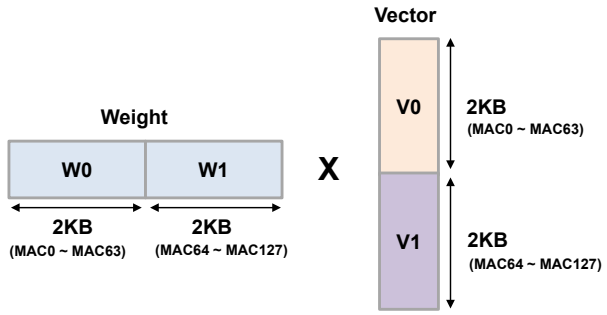


AiM: Accumulation Operation

- Up to 64 MAC commands between ACT
  - 1 MAC every 1ns (t_{CCDS})
  - 64 x 256 bits = 2 KB (row size)

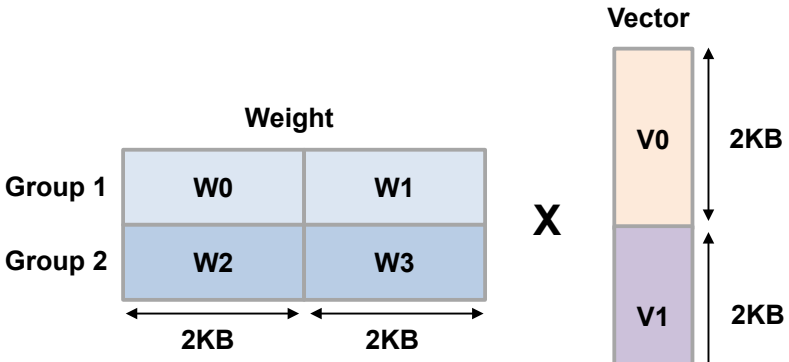
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

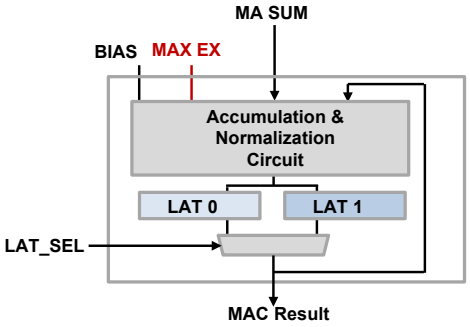




AiM: Multi-latch Operation

- Save writes with selectable latches

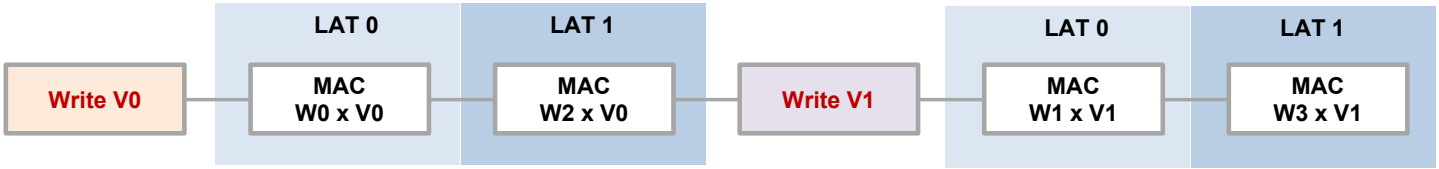




Accumulator with Two Selectable Latches Storing MAC Results

Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

Operation Sequence without Multi-Latch

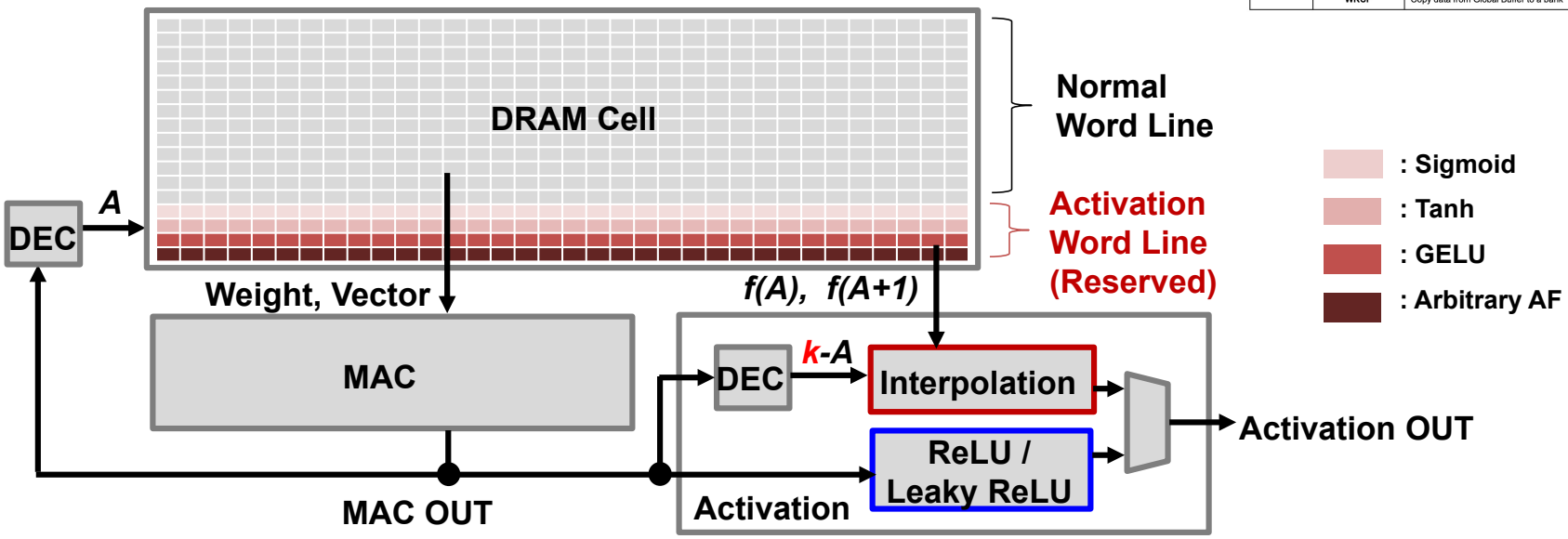


Operation Sequence with Multi-Latch

AiM: Activation Functions (I)

- Two types of activation functions
  - Calculation: ReLU, Leaky ReLU
  - With LUT: Sigmoid, GELU, Tanh, arbitrary AF

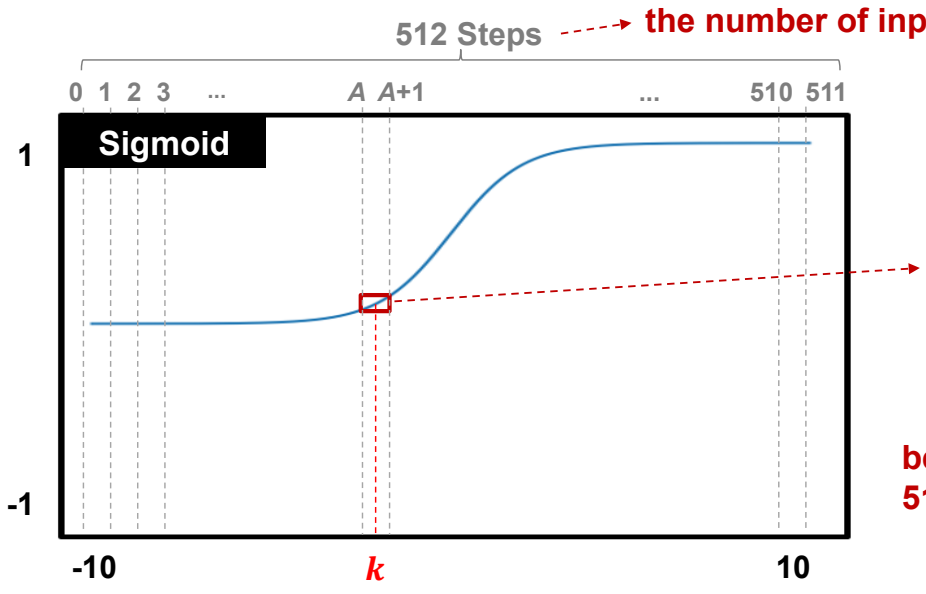
Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank

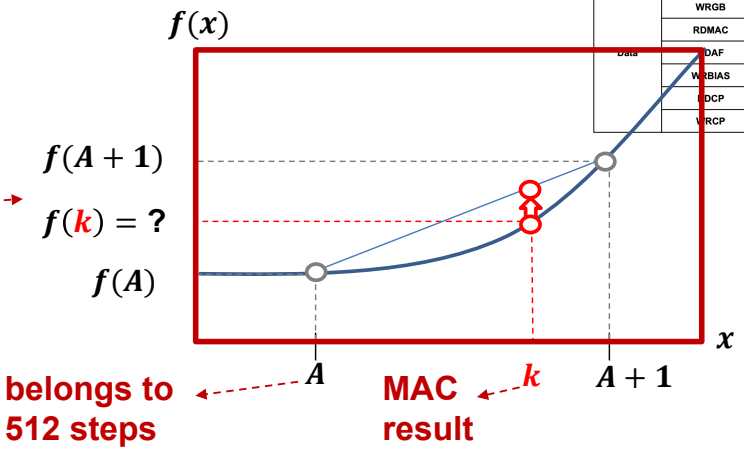


AiM: Activation Functions (II)

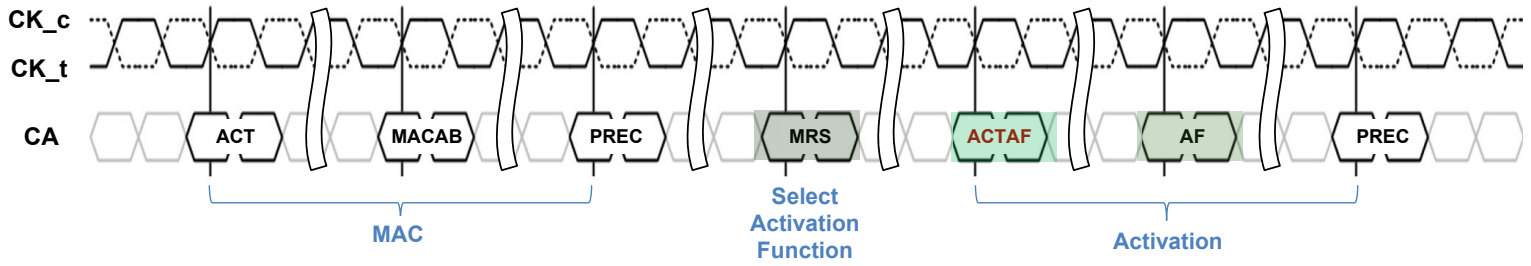
- LUT (512 values) + Linear interpolation
  - Mode set register (MSR) to select AF

Type	CMD	Description
Bank Activation	ACT4, ACT16	Activate four/sixteen banks in parallel
	ACTAF4, ACTAF16	Activate rows storing activation function LUTs in four/sixteen banks in parallel
Compute	MACSB, MAC4B, MACAB	Perform MAC in one/four/sixteen banks in parallel
	AF	Compute activation function in all banks
	EWMUL	Perform element-wise multiplication
Data	WRBK	Write to all activated banks in parallel
	WRGB	Write to Global Buffer
	RDMAC	Read from MAC result register
	RDAF	Read from activation function result register
	WRBIAS	Write bias to MAC result register
	RDCP	Copy data from a bank to Global Buffer
	WRCP	Copy data from Global Buffer to a bank





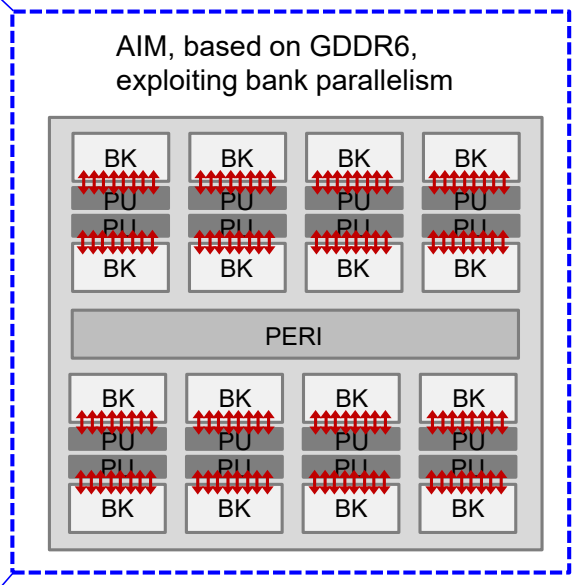
$$f(k) \fallingdotseq f(A) + \frac{k - A}{(A+1) - A} \times (f(A+1) - f(A))$$



AiM: Key Feature Summary

- Comparison table

	[1]	UPMEM PIM [2]	FIMDRAM [3, 7]	This work
DRAM Type	LPDDR4	DDR4	HBM2	GDDR6
Process	20nm	2x nm	20 nm	1y nm
Memory Density	8GB/chip (8H 8Gb mono die)	8GB/DIMM	6GB/cube (Buffer die + 4H 4Gb core-die with + 4H 8Gb core-die)	8Gb/chip (4Gb DDP)
Data Rate	3.2Gbps	2.4Gbps	2.4Gbps	16Gbps
Bandwidth	25.6GB/s per chip	19.2GB/s per DIMM	307GB/s per cube	64GB/s per chip
# of Processing Unit (PU)	2048 per chip (256 per die)	128 per DIMM (8 per chip)	128 per cube (32 per core-die)	32 per chip (16 per die)
Processing Operation Speed	250MHz	500MHz	300MHz	1GHz
1 PU Throughput	2 GOPS (250MHz x 8byte)	4 GOPS (500MHz x 8byte)	9.6 GFLOPS (300MHz x 32byte)	32 GFLOPS (1GHz x 32byte)
Total Throughput (1 PU Throughput x # of PU)	0.5 TOPS per chip (2 GOPS x 256)	0.5 TOPS per DIMM (4 GOPS x 128)	1.2 TFLOPS per cube (9.6 GFLOPS x 128)	1 TFLOPS per chip (32 GFLOPS x 32)
Operation precision	INT8	INT8	FP16	BF16
Supported Activation Functions	-	-	ReLU	Sigmoid, Tanh, GELU, ReLU, Leaky ReLU and Arbitrary AF



[1] H. Shin, et al., IEEE TCADICS 2018,
[2] F. Devaux, IEEE Hot Chips Symp. 2019,
[3] Y.-C. Kwon et al., ISSCC 2021

Upcoming Lectures

- More real-world PIM architectures
- Programming PIM systems
- More on workload characterization for PIM suitability
- PUM architectures and prototypes

P&S Processing-in-Memory

Real-World Processing-in-Memory Architectures: SK Hynix Accelerator-in-Memory

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich

Fall 2022

15 November 2022